Approximate Inference: Loopy BP & Variational Methods

Excerpted from:
Learning in Graphical Models

Prof. Alexander Ihler

BREN:ICS
INFORMATION AND COMPUTER SCIENCES

UNIVERSITY of CALIFORNIA IRVINE

Common inference query tasks

- Model

$$p(\mathbf{x}) \propto f(\mathbf{x}) = \prod_{\alpha} \psi_\alpha(\mathbf{x}_\alpha)$$

"α" are sets of variables (function arguments)
Factors may be conditional probabilities (BNs)
or un-normalized positive factors (MRFs, FGs)

- Maximization
  - Compute the most probable state, or its value

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \prod_{\alpha} \psi_\alpha(\mathbf{x}_\alpha) \qquad f(\mathbf{x}^*) = \max_{\mathbf{x}} \prod_{\alpha} \psi_\alpha(\mathbf{x}_\alpha)$$

- Summation
  - Compute marginal probabilities or normalization constant

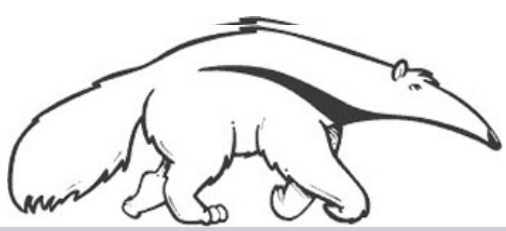
"partition function"

$$p(x_i) = \frac{1}{Z} \sum_{\mathbf{x} \setminus x_i} \prod_{\alpha} \psi_\alpha(\mathbf{x}_\alpha) \quad \text{and} \quad Z = \sum_{\mathbf{x}} \prod_{\alpha} \psi_\alpha(\mathbf{x}_\alpha)$$

Inference: Loopy Belief Propagation

Learning in Graphical Models

Prof. Alexander Ihler



BREN:ICS
INFORMATION AND COMPUTER SCIENCES

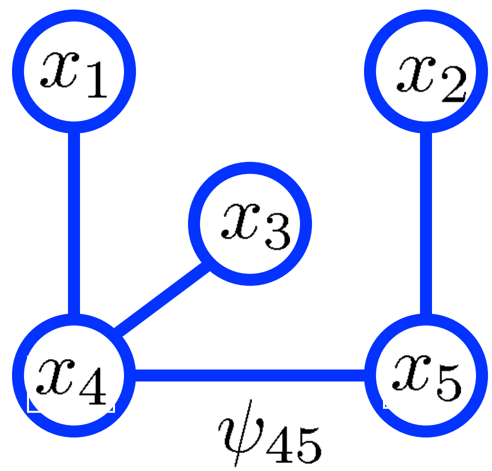
UNIVERSITY of CALIFORNIA IRVINE

Variable Elimination in Trees

(Use distributive rule to calculate efficiently:)

$$p(x_5) \propto \sum_{x_1 \ldots x_4} \psi_{14}\psi_{25}\psi_{34}\psi_{45}$$

$$\propto \sum_{x_2 \ldots x_4} \psi_{25}\psi_{34}\psi_{45} \sum_{x_1} \psi_{14}$$



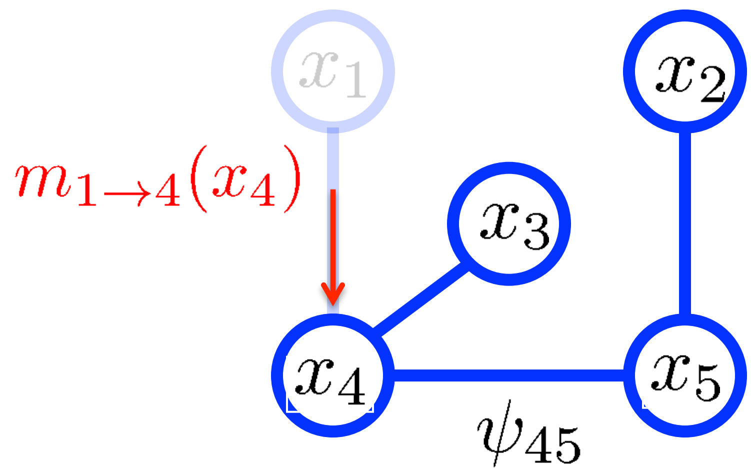
Variable Elimination in Trees

(Use distributive rule to calculate efficiently:)

$$p(x_5) \propto \sum_{x_1 \ldots x_4} \psi_{14}\psi_{25}\psi_{34}\psi_{45}$$

$$\propto \sum_{x_2 \ldots x_4} \psi_{25}\psi_{34}\psi_{45} \sum_{x_1} \psi_{14}$$

$$\propto \sum_{x_3 \ldots x_4} \psi_{34}\psi_{45} \; m_{1\to 4}(x_4) \; \sum_{x_2} \psi_{25}$$



Variable Elimination in Trees

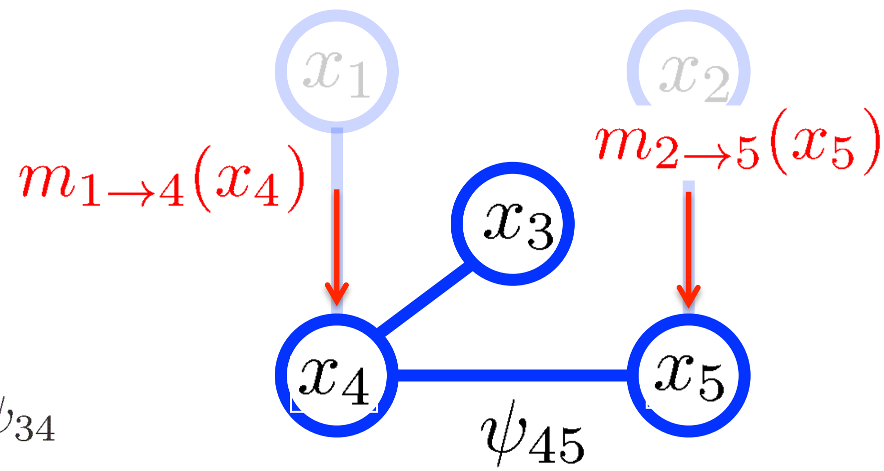
(Use distributive rule to calculate efficiently:)

$$p(x_5) \propto \sum_{x_1 \ldots x_4} \psi_{14}\psi_{25}\psi_{34}\psi_{45}$$

$$\propto \sum_{x_2 \ldots x_4} \psi_{25}\psi_{34}\psi_{45} \sum_{x_1} \psi_{14}$$

$$\propto \sum_{x_3 \ldots x_4} \psi_{34}\psi_{45}\; m_{1\to4}(x_4) \sum_{x_2} \psi_{25}$$

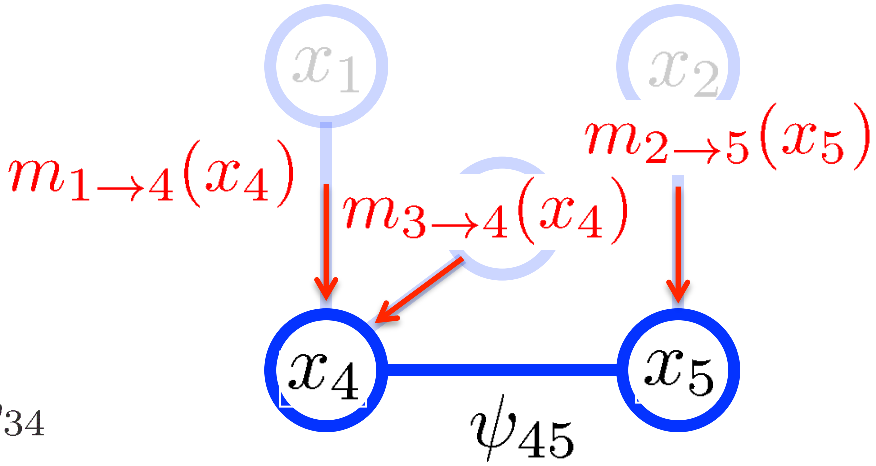
$$\propto \sum_{x_4} \psi_{45}\; m_{1\to4}(x_4)\; m_{2\to5}(x_5) \sum_{x_3} \psi_{34}$$



Variable Elimination in Trees

(Use distributive rule to calculate efficiently:)

$$
\begin{aligned}
p(x_5) &\propto \sum_{x_1 \ldots x_4} \psi_{14}\psi_{25}\psi_{34}\psi_{45} \\
&\propto \sum_{x_2 \ldots x_4} \psi_{25}\psi_{34}\psi_{45} \sum_{x_1} \psi_{14} \\
&\propto \sum_{x_3 \ldots x_4} \psi_{34}\psi_{45} \; m_{1\to4}(x_4) \sum_{x_2} \psi_{25} \\
&\propto \sum_{x_4} \psi_{45} \; m_{1\to4}(x_4) \; m_{2\to5}(x_5) \sum_{x_3} \psi_{34} \\
&\propto m_{2\to5}(x_5) \sum_{x_4} \psi_{45} \; m_{1\to4}(x_4) \; m_{3\to4}(x_4)
\end{aligned}
$$



Variable Elimination in Trees

(Use distributive rule to calculate efficiently:)

$$p(x_5) \propto \sum_{x_1 \dots x_4} \psi_{14}\psi_{25}\psi_{34}\psi_{45}$$

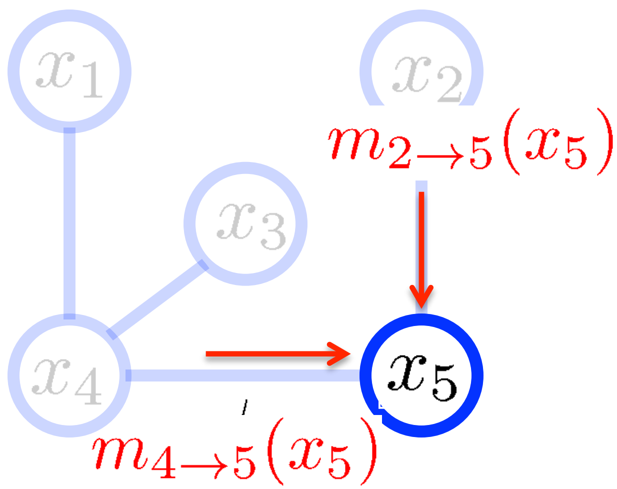
$$\propto \sum_{x_2 \dots x_4} \psi_{25}\psi_{34}\psi_{45} \sum_{x_1} \psi_{14}$$

$$\propto \sum_{x_3 \dots x_4} \psi_{34}\psi_{45}\ m_{1\to4}(x_4)\ \sum_{x_2} \psi_{25}$$

$$\propto \sum_{x_4} \psi_{45}\ m_{1\to4}(x_4)\ m_{2\to5}(x_5)\ \sum_{x_3} \psi_{34}$$

$$\propto m_{2\to5}(x_5)\ \sum_{x_4} \psi_{45}\ m_{1\to4}(x_4)\ m_{3\to4}(x_4)$$

$$\propto m_{2\to5}(x_5)\ m_{4\to5}(x_5)$$



At each step, calculate message:

$$m_{j\to i}(x_i) = \sum_{x_j} \psi_{ij}\ \psi_i \prod_{k\neq i} m_{k\to j}(x_j)$$

Variable elimination in trees

(Use distributive rule to calculate efficiently:)

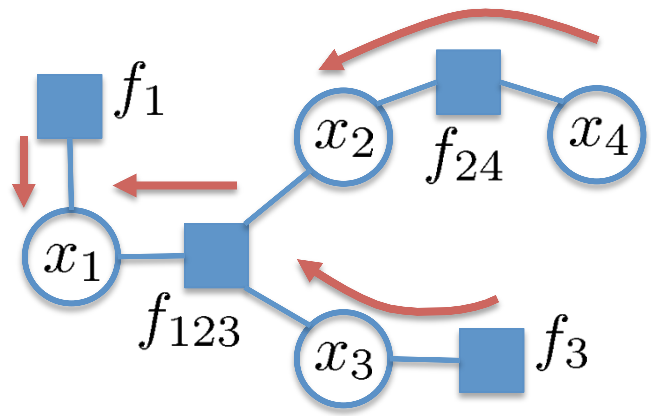
$$p(x_1) \propto \sum_{x_2 \dots x_4} f_1(x_1)\, f_{123}(x_1,x_2,x_3)\, f_{24}(x_2,x_4)\, f_3(x_3)$$

$$\propto \sum_{x_2,x_3} f_1\, f_{123}\, f_3 \sum_{x_4} f_{24}$$

$$\propto \sum_{x_2,x_3} f_1\, f_{123}\, f_3\, m_{24\to2}(x_2)$$

$$\propto f_1 \sum_{x_2,x_3} f_{123}\, m_{3\to3}(x_3)\, m_{24\to2}(x_2)$$

$$\propto f_1\, m_{123\to1}(x_1)$$



Calculating messages:

$$m_{i\to\alpha}(x_i) \propto \prod_{\beta\neq\alpha} m_{\beta\to i}(x_i)$$

$$m_{\alpha\to i}(x_i) \propto \sum_{x_\alpha \setminus x_i} f_\alpha(x_\alpha) \prod_{j\neq i} m_{j\to\alpha}(x_j)$$

Variable elimination in trees

Computing the marginal at another variable reuses much of the computation!

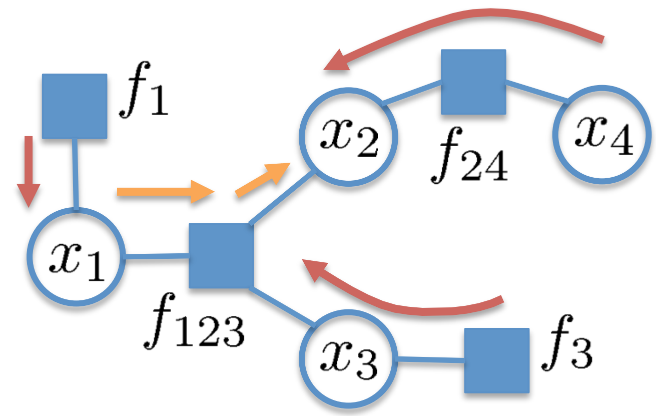
$$p(x_2) \propto \sum_{x_1,x_3,x_4} f_1(x_1)\, f_{123}(x_1,x_2,x_3)\, f_{24}(x_2,x_4)\, f_3(x_3)$$

$$\propto \sum_{x_1,x_3} f_1\, f_{123}\, f_3 \sum_{x_4} f_{24}$$

$$\propto \sum_{x_1,x_3} f_1\, f_{123}\, f_3\, m_{24\to 2}(x_2)$$

$$\propto \sum_{x_1,x_3} f_1\, f_{123}\, m_{3\to 3}(x_3)\, m_{24\to 2}(x_2)$$

$$\propto m_{123\to 2}(x_2)\, m_{24\to 2}(x_2)$$



Calculating messages:

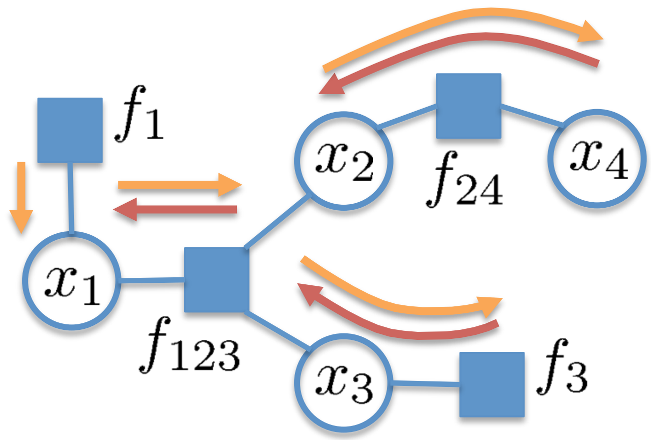
$$m_{i\to\alpha}(x_i) \propto \prod_{\beta\neq\alpha} m_{\beta\to i}(x_i)$$

$$m_{\alpha\to i}(x_i) \propto \sum_{x_\alpha \setminus x_i} f_\alpha(x_\alpha) \prod_{j\neq i} m_{j\to\alpha}(x_j)$$

Variable elimination in trees

Computing all marginal probabilities:

- Two-pass algorithm
  - Pass messages upward to a root
  - Pass messages back downward
  - Messages summarize marginalization of sub-model rooted at that node



- Use messages to compute marginals
- Can also update all messages in parallel, until converged

Calculating messages:

$$m_{i\to\alpha}(x_i) \propto \prod_{\beta\neq\alpha} m_{\beta\to i}(x_i)$$

$$m_{\alpha\to i}(x_i) \propto \sum_{x_\alpha\setminus x_i} f_\alpha(x_\alpha) \prod_{j\neq i} m_{j\to\alpha}(x_j)$$

Calculating marginals:

$$p(x_i) \propto \prod_{\alpha\ni i} m_{\alpha\to i}(x_i)$$

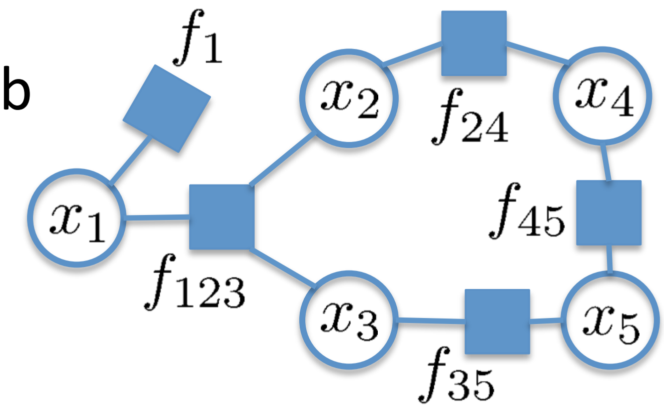
$$p(x_\alpha) \propto f_\alpha(x_\alpha) \prod_{i\in\alpha} m_{i\to\alpha}(x_i)$$

Loopy belief propagation

- Apply the same local updates in arbitrary structure
  - More precisely, often called the "sum-product" algorithm

- Resulting algorithm computes "beliefs" b
  - May not converge
  - Initialization & schedule can matter
  - Is approximate: $b(x_i) \approx p(x_i)$
  - But, is often pretty good in practice

    (quality depends on how "tree-like" the model is)



Calculating messages:

$$m_{i\to\alpha}(x_i) \propto \prod_{\beta\neq\alpha} m_{\beta\to i}(x_i)$$

$$m_{\alpha\to i}(x_i) \propto \sum_{x_\alpha\setminus x_i} f_\alpha(x_\alpha) \prod_{j\neq i} m_{j\to\alpha}(x_j)$$

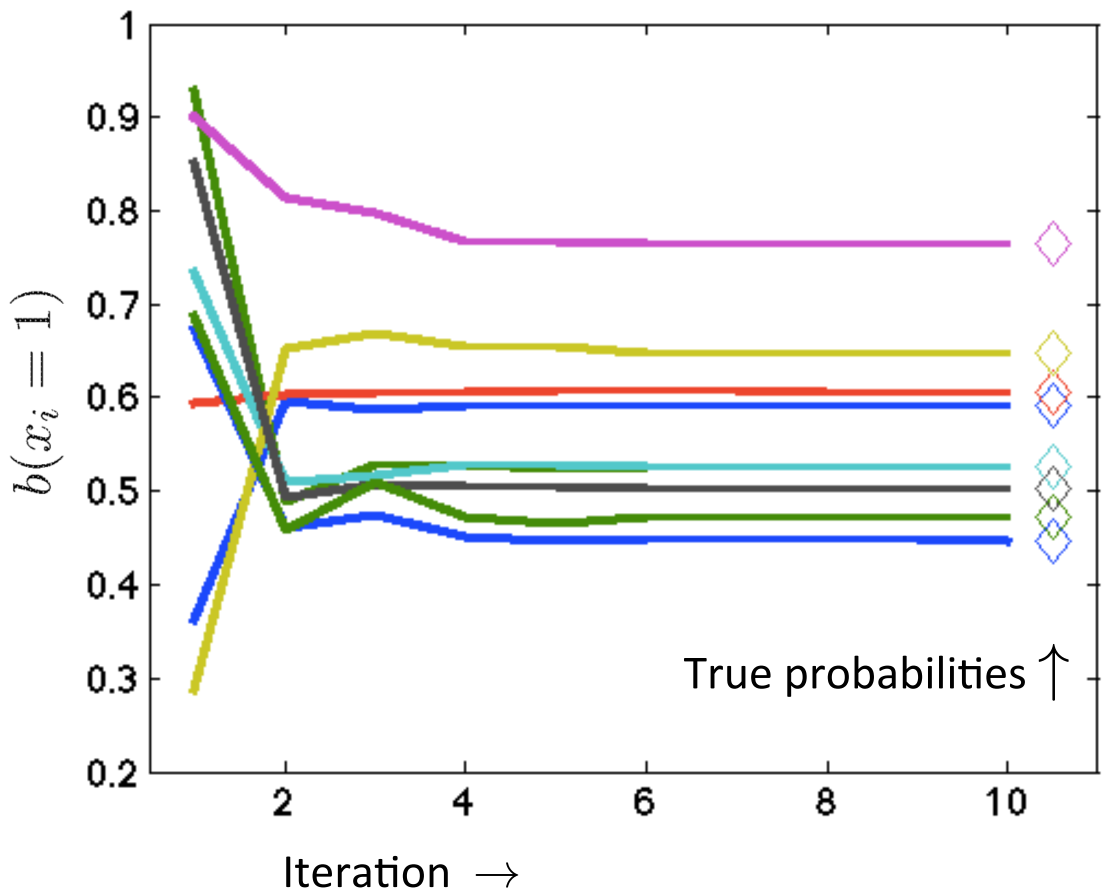
Calculating marginals:

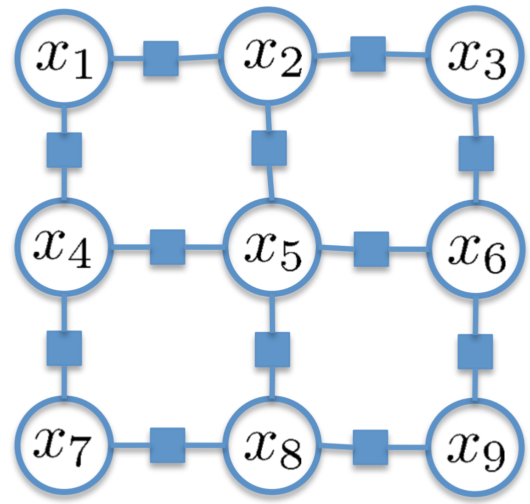
$$b(x_i) \propto \prod_{\alpha\ni i} m_{\alpha\to i}(x_i)$$

$$b(x_\alpha) \propto f_\alpha(x_\alpha) \prod_{i\in\alpha} m_{i\to\alpha}(x_i)$$

Example: Ising model

- Log-factors $\theta_i, \theta_{ij} \sim U[-0.5, 0.5]$

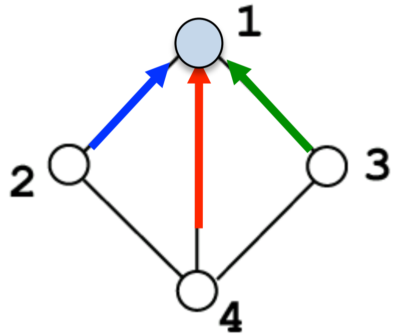


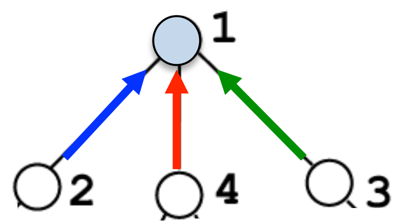


Computation trees

(Weiss & Freeman '99),
and many others

- Analyze the behavior of loopy BP
  - Tree-structured "unrolling" of loopy graph

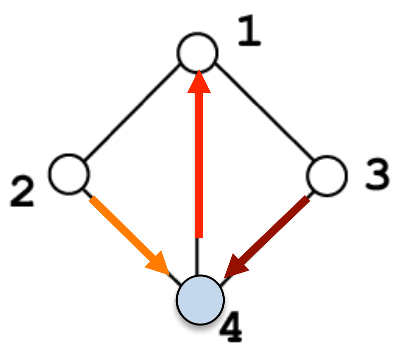


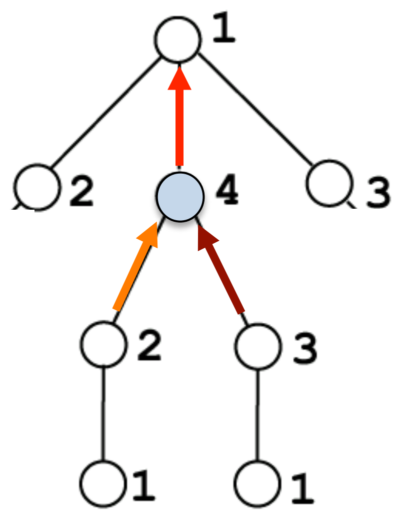


Computation trees

(Weiss & Freeman '99), and many others

- Analyze the behavior of loopy BP
  - Tree-structured "unrolling" of loopy graph

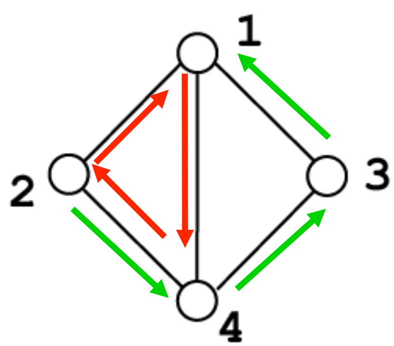


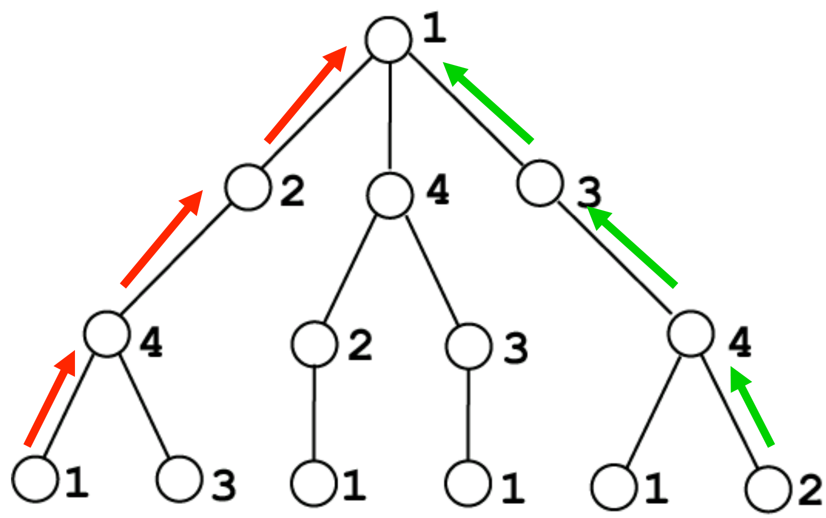


Computation trees

(Weiss & Freeman '99), and many others

- Analyze the behavior of loopy BP
  - Tree-structured "unrolling" of loopy graph
  - Contains all length-N "forward" paths

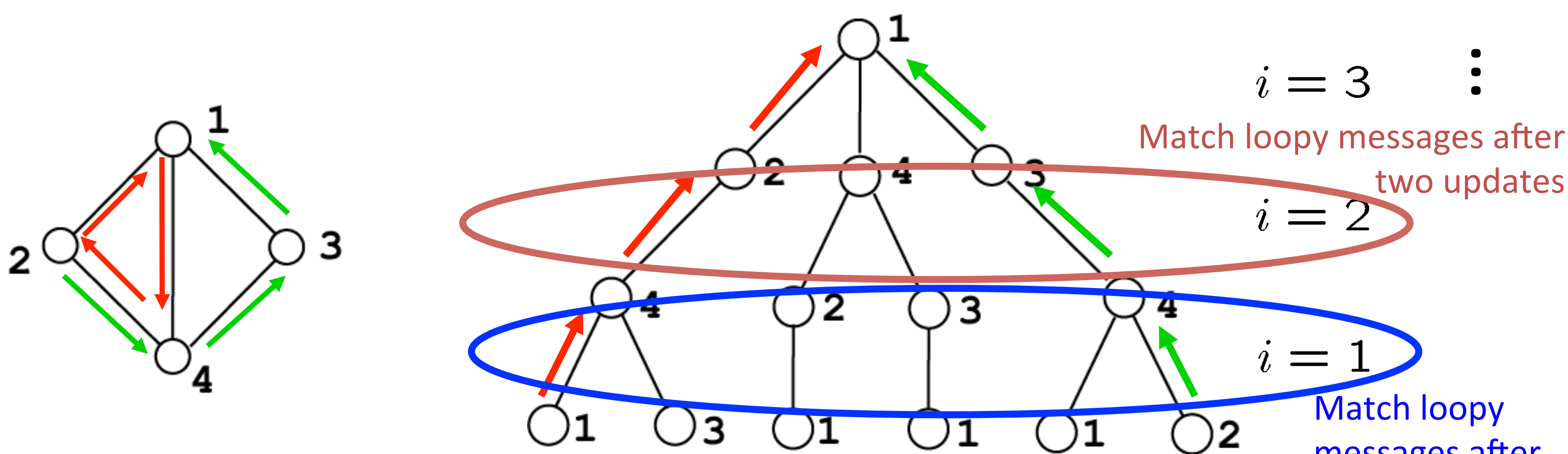




Computation trees

(Weiss & Freeman '99), and many others

- Analyze the behavior of loopy BP
  - Tree-structured "unrolling" of loopy graph
  - Contains all length-N "forward" paths
  - At the root, equivalence between
    - N iterations of loopy BP (all messages update in parallel)
    - N steps of BP on computation tree (upward pass)

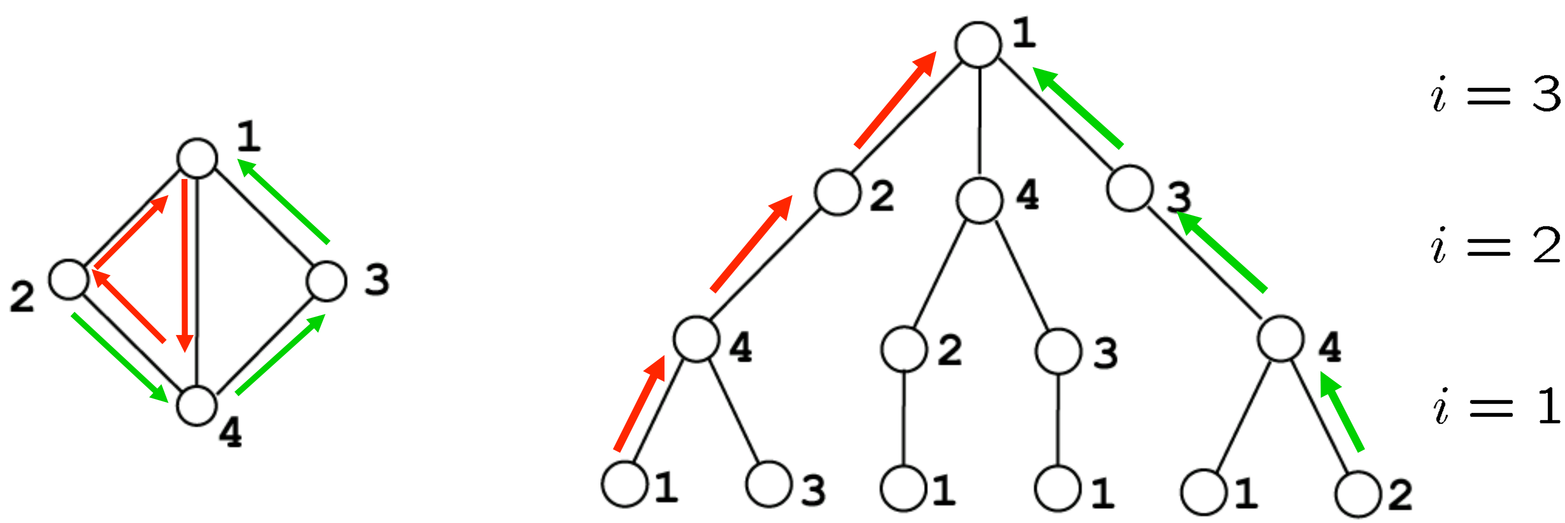


Initialize leaves to match initial messages of loopy BP

Computation trees

(Tatikonda & Jordan 2003;
Ihler et al. 2005;
Ihler 2007; etc.)

- When does loopy BP converge / work well?
  - Converges if root "decouples" from (growing set of) leaves
  - Accurate if the computation tree is "close" to the original distribution
  - Examples of how this can occur:
    - Some nodes have strong evidence ("nearly observed")
    - Edges forming cycles are weak ("fast mixing")

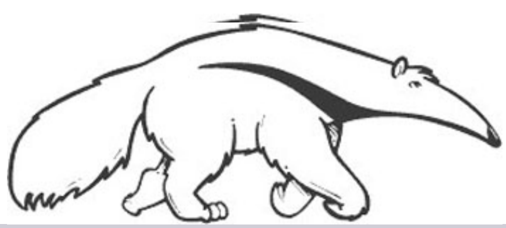


Initialize leaves to match initial messages of loopy BP

Linear Programming Relaxations & Dual Decomposition

Learning in Graphical Models

Prof. Alexander Ihler



BREN:ICS
INFORMATION AND COMPUTER SCIENCES

UNIVERSITY of CALIFORNIA IRVINE

$$p(\mathbf{x}) \propto f(\mathbf{x}) = \prod_{\alpha} \psi_\alpha(\mathbf{x}_\alpha)$$

Integer Linear Program

- Consider the MAP inference problem

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \prod_{\alpha} \psi_\alpha(\mathbf{x}_\alpha) \qquad\qquad f(\mathbf{x}^*) = \max_{\mathbf{x}} \prod_{\alpha} \psi_\alpha(\mathbf{x}_\alpha)$$

  - Let's assume pairwise, so $\alpha \in \{\, i \,\} \cup \{\, (i,j) \,\}$

- Helpful to write f(x) as overcomplete exponential family

$$f(x) = \exp\left[\sum_{i,k} \theta_{i;k}\, u_{i;k}(x) + \sum_{i,j,k,l} \theta_{ij;kl}\, u_{ij;kl}(x)\right]$$

$$= \exp\Big[\theta \cdot u(x)\Big]$$

Features:

$$u_{i;k}(x) = \delta[\, x_i = k \,]$$

$$u_{ij;kl}(x) = \delta[\, x_i = k \;\&\; x_j = l \,]$$

Parameter & feature vectors:

$$\theta = [\ldots \quad \theta_{i;k} \quad \ldots \quad \theta_{ij;kl} \quad \ldots]$$

$$u(x) = [\ldots u_{i;k}(x) \ldots u_{ij;kl}(x) \ldots]$$

Integer Linear Program

- Then, MAP is an integer linear program:

$$\log f(x^*) = \max_u \left[\sum_{i,k} \theta_{i;k}\, u_{i;k} + \sum_{i,j,k,l} \theta_{ij;kl}\, u_{ij;kl} \right]$$

(Linear objective)

where

$$u_{i;k} \in \{0,1\}$$
$$u_{ij;kl} \in \{0,1\}$$

u(x) are indicator functions

(Integrality)

$$\sum_k u_{i;k} = 1$$
$$\sum_{k,l} u_{ij;kl} = 1$$

Only one indicator function is on per clique

(Linear constraints)

$$\sum_l u_{ij;kl} = u_{i;k}$$
$$\sum_k u_{ij;kl} = u_{j;l}$$

Configuration of (x_i, x_j) is consistent with x_i and x_j

Equivalent to

$$u_{ij;kl} = 1 \Rightarrow u_{i;k} = u_{j;l} = 1$$

Linear Programming Relaxation

- Relaxing the integrality constraints gives a linear program:

$$\log f(x^*) \leq \max_{\mu} \left[\sum_{i,k} \theta_{i;k}\, \mu_{i;k} + \sum_{i,j,k,l} \theta_{ij;kl}\, \mu_{ij;kl} \right]$$

(Linear objective)

where

$$\mu_{i;k} \in [0,1]$$
$$\mu_{ij;kl} \in [0,1]$$

Relax integrality {0,1} to any real value in [0,1]

(Use μ to represent relaxed indicators)

$$\sum_{k} \mu_{i;k} = 1$$
$$\sum_{k,l} \mu_{ij;kl} = 1$$

Only one indicator function is on per clique

(Linear constraints)

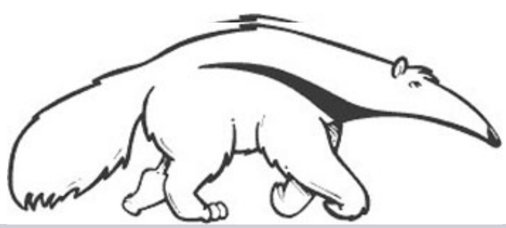
$$\sum_{l} \mu_{ij;kl} = \mu_{i;k}$$
$$\sum_{k} \mu_{ij;kl} = \mu_{j;l}$$

Configuration of (x_i, x_j) is consistent with x_i and x_j

Variational Methods

Learning in Graphical Models

Prof. Alexander Ihler



BREN:ICS
INFORMATION AND COMPUTER SCIENCES

UNIVERSITY of CALIFORNIA IRVINE

Variational algorithms

$$f(x) = \prod_\alpha \psi_\alpha(x_\alpha)$$

- Replace “elimination” with optimization over distributions

  Elimination: $x^* = \arg\max_x \; f(x) \qquad f^* = f(x^*) = \max_x \; f(x)$

  Variational: $\log f(x^*) = \max_{q \in \mathbb{P}} \; \mathbb{E}_q[\log f(x)]$

- Optimal q puts all mass on optimal value(s) of x

  $$q^*(x) = \delta(x = x^*)$$

  – Mass on any non-optimal configuration reduces the average value

- Exponential family f(x): $f(x) = \exp(\sum_i \theta_i \; u_i(x)\,)$

$$\log f(x^*) = \max_{q \in \mathbb{P}} \sum_i \theta_i \mathbb{E}_q[u_i(x)] = \max_{\mu \in \mathcal{M}} \sum_i \theta_i \, \mu_i$$

(M: set of all moments achievable by some q)

Marginal polytope view

- What moments are achievable by some q?
  - A convex set on the moments μ

Overcomplete exponential family:

$x_1 \in \{0,1\}\ x_2 \in \{0,1\}$

$$u_{12;kl}(x_1, x_2) = \delta[x_1 = k, x_2 = l]$$

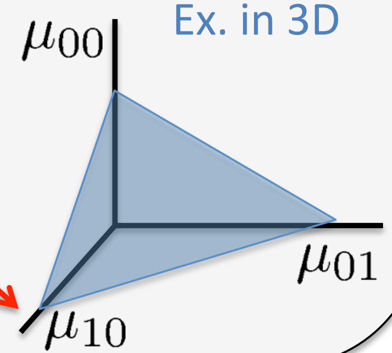
$$\mu_{12;kl} = \mathbb{E}_q(\,u_{12;kl}\,) = q_{kl} \in [0,1]$$

$$q(x_1, x_2) = \begin{bmatrix} q_{00} & q_{01} \\ q_{10} & q_{11} \end{bmatrix}$$

$$\sum_{kl} \mu_{12;kl} = 1$$

M = unit simplex

Ex. in 3D



Why? Discrete vector x:
q(x) puts some weight on each x
M = set of all valid q(x)
= the convex combination of x's

Vertices, e.g. (0,0,1):
distribution is an
indicator f'n,
$\delta(x_1 = 1, x_2 = 0)$

Marginal polytope view

- What moments are achievable by some q?
  - A convex set on the moments μ

Ising model:

$x_1 \in \{0,1\}$ $x_2 \in \{0,1\}$

$$u_1(x_1) = \delta[x_1 = 1]$$
$$u_2(x_2) = \delta[x_2 = 1]$$
$$u_{12}(x_1, x_2) = \delta[x_1 = 1, x_2 = 1]$$

$$q(x_1, x_2) = \begin{bmatrix} q_{00} & q_{01} \\ q_{10} & q_{11} \end{bmatrix}$$

$$\mu_{12} = \mathbb{E}_q(\, u_{12} \,) = q_{11} \in [0, 1]$$
$$\mu_1 = \mathbb{E}_q(\, u_1 \,) = q_{10} + q_{11} \in [\mu_{12}, 1]$$
$$\mu_2 = \mathbb{E}_q(\, u_2 \,) = q_{01} + q_{11} \in [\mu_{12}, 1]$$

$$\mu_1 + \mu_2 - \mu_{12} = q_{10} + q_{01} + q_{11} \leq 1$$

M is a polytope
(conjuction of hyperplanes)

Vertices are indicator f'ns

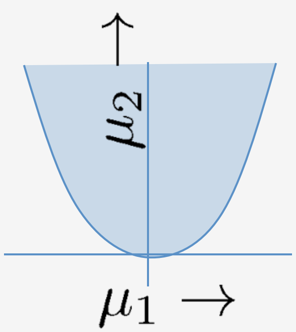
Marginal polytope view

- What moments are achievable by some q?
  - A convex set on the moments μ

Gaussian model: $x \in \mathbb{R}$

$$u_1(x) = x$$
$$u_2(x) = x^2$$

$$\mu_1 = \mathbb{E}_q(\, x\,)$$
$$\mu_2 = \mathbb{E}_q(\, x^2\,) = \underbrace{\mathrm{Var}_q[x]}_{\geq 0} + \underbrace{(\mathbb{E}_q[x])^2}_{= (\mu_1)^2}$$



M = convex; quadratic shape

Connection to linear programming

- For a discrete optimization, a variational form is

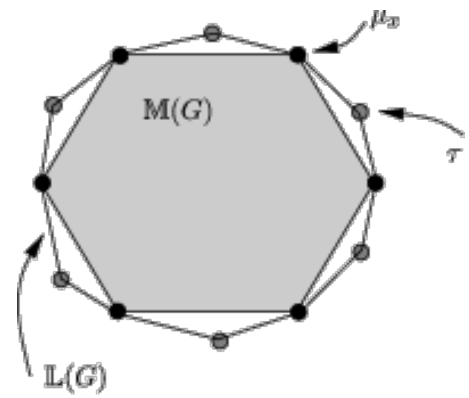
$$\log f(x^*) = \max_{q \in \mathbb{P}} \mathbb{E}[\log f(x)] = \max_{\mu \in \mathcal{M}} \Big[\sum_{i,k} \theta_{i;k}\, \mu_{i;k} + \sum_{i,j,k,l} \theta_{ij;kl}\, \mu_{ij;kl} \Big]$$

  - However, there are *many* constraints in M…

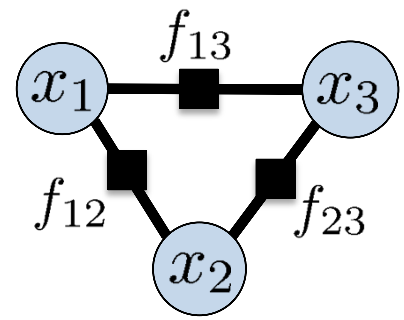
- We can elect to only enforce some of those constraints, e.g., the *local* polytope:

$$\mu \in \mathbb{L} : \quad \begin{array}{ccc} \mu_{i;k} \in [0,1] & \sum_k \mu_{i;k} = 1 & \sum_l \mu_{ij;kl} = \mu_{i;k} \\ \mu_{ij;kl} \in [0,1] & \sum_{k,l} \mu_{ij;kl} = 1 & \sum_k \mu_{ij;kl} = \mu_{j;l} \end{array}$$

  - Gives exactly the LP relaxation from before
  - Any additional constraints will tighten the relaxation



The local polytope



- Local polytope enforces only some constraints:

  Example:

$$\mu_1 = \mu_2 = \mu_3 \quad \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

$$\mu_{12} \quad \begin{array}{cc} & x_2 \\ x_1 & \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \end{array} \quad (x_1 = x_2)$$

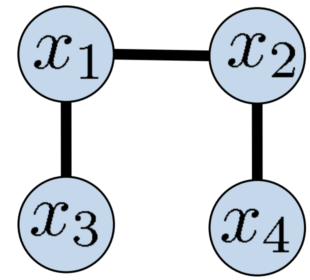
$$\mu_{13} \quad \begin{array}{cc} & x_3 \\ x_1 & \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \end{array} \quad (x_1 = x_3)$$

$$\mu_{23} \quad \begin{array}{cc} & x_3 \\ x_2 & \begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix} \end{array} \quad (x_2 \neq x_3)$$

  – All pairwise moments are consistent with singleton moments
  – But, no consistent distribution q(x) exists

- Example illustrates the connection to arc-consistency in CSPs
  – Enforces a local consistency, but may not be globally consistent
    (similar, less "extreme" example without hard zeros)

Local polytope for trees



- Suppose we have arbitrary moments $\mu \in \mathbb{L}$:

$$0 \le \mu_1(k) \le 1 \qquad 0 \le \mu_{12}(k,l) \le 1 \qquad \sum_l \mu_{12}(k,l) = \mu_1(k)$$
$$0 \le \mu_2(k) \le 1 \qquad 0 \le \mu_{13}(k,l) \le 1 \qquad \sum_k \mu_{12}(k,l) = \mu_2(l)$$
$$0 \le \mu_3(k) \le 1 \qquad 0 \le \mu_{24}(k,l) \le 1 \qquad \sum_l \mu_{13}(k,l) = \mu_1(k)$$
$$0 \le \mu_4(k) \le 1 \qquad \sum_k \mu_{13}(k,l) = \mu_3(l)$$
$$\vdots$$

  defined (only) on the tree-structured graph G

- (Constructive) example that achieves these moments:

$$q(x_1) = \mu_1(x_1) \quad q(x_2|x_1) = \frac{\mu_{12}(x_1,x_2)}{\mu_1(x_1)} \quad q(x_3|x_1) = \frac{\mu_{13}(x_1,x_3)}{\mu_1(x_1)} \quad \ldots$$

  - Similar to closed-form MLE for trees
  - Root to leaves; no possible "contradicting" moments
  - Alt., more symmetric form: $q(x) = \prod_i \mu_i(x_i) \prod_{ij} \frac{\mu_{ij}(x_i,x_j)}{\mu_i(x_i)\mu_j(x_j)}$

Duality and the log-partition function

- Represent LPF as optimization over distributions

$$\log Z = \max_{q \in \mathbb{P}} \; \mathbb{E}_q[\log f(x)] + H(x\,;\,q)$$

(maximum: $q = p$)

- Two problems:
  - Constraint set is too complex (as in LP form)
  - Entropy defined in terms of q, not available

- Proof: $$D(q\|p) = \sum_x q(x) \log \left[\frac{q(x)}{\frac{1}{Z} f(x)} \right]$$

(Kullback–Leibler divergence)

$$= -H(x\,;\,q) - \mathbb{E}_q[\log f(x)] + \log Z$$

$$\Rightarrow \log Z \geq \mathbb{E}_q[\log f(x)] + H(x\,;\,q)$$

equal iff $p = q$

Variational approximations

- Replace q ∈ P and H(q) with simpler approximations

$$\log p(x^*) = \max_{q \in \mathbb{P}} \; \mathbb{E}_q[\log \psi(x)]$$

$$\log Z = \max_{q \in \mathbb{P}} \; \mathbb{E}_q[\log \psi(x)] + H(x\,;\,q)$$

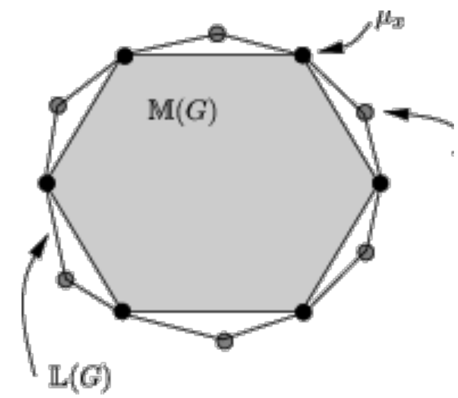
- Algorithms & their properties:

	Method	distributions	entropy	value
Max:	Linear programming	$q \in \mathbb{L} \supseteq \mathbb{P}$	n/a	$\hat{p}_{lp} \geq p(x^*)$
Sum:	Mean field	$\{q = \prod q_i(x_i)\} \subseteq \mathbb{P}$	exact	$Z_{mf} \leq Z$
	Belief propagation	$q \in \mathbb{L} \supseteq \mathbb{P}$	$H_\beta \approx H(q)$	$Z_\beta \approx Z$
	Tree-reweighted	$q \in \mathbb{L} \supseteq \mathbb{P}$	$H_{tr} \geq H(q)$	$Z_{tr} \geq Z$

Approximating the marginal polytope

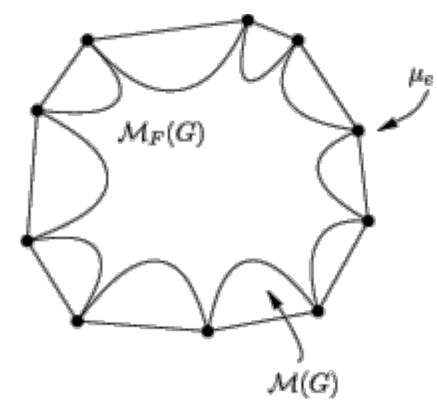
Outer bound: locally consistent polytope

$$\mathcal{L}(G) = \{b_i, b_{ij} | \sum_{x_i} b_{ij}(x_i, x_j) = b_j(x_j), \sum_{x_j} b_j(x_j) = 1, b_{ij} \geq 0\}$$



Inner bound: "naïve mean field" polytope

$$\mathcal{M}_{mf}(G) = \{q | q(x) = \prod_i b_i(x_i) \text{ for some } b_i\}$$



Naïve Mean Field

- Restrict q(x) to a domain where H(q) is easy:
  - Naïve mean field: q(x) is fully independent

$$q(x) = \prod_i q_i(x_i) \quad \Rightarrow \quad H(q) = \sum_i H(q_i)$$

- Optimizing the bound via coordinate ascent:

$$\max_{q_i \in \mathrm{MF}} \quad \mathbb{E}_q[\theta(x)] + H(q) = \mathbb{E}_q\Big[\sum_{\alpha \ni i} \theta_\alpha(x_\alpha)\Big] + H(q_i) + \mathrm{const}$$

$$= \log g(x_i) + H(q_i)$$

$$= D(\, q_i \,\|\, g_i \,)$$

$$\log g_i(x_i) = \mathbb{E}_{q_{\neg i}}\Big[\sum_{\alpha \ni i} \theta_\alpha(x_\alpha)\Big]$$

$$q_{\neg i}(x) = \prod_{j \neq i} q_j(x_j)$$

Coordinate update:

$$\Rightarrow \quad q_i(x_i) \propto \exp\Big[\, \mathbb{E}_{q_{\neg i}}\Big[\sum_{\alpha \ni i} \theta_\alpha(x_\alpha)\Big]\Big]$$

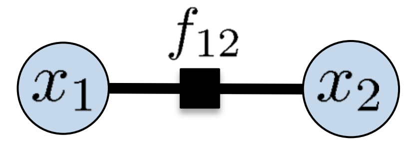
Mean Field Example

```
th12 = factor([1 2], [0 1 ; 0.5  3 ]);% theta(x) = ...
f12 = exp(th12);                       % f(x) = exp( theta(x) )
lnZ = log(sum( f12 ));                 % ln Zf = 3.237

q1 = factor(1, [.5 .5]);         % initialize factored q(x)
q2 = factor(2, [.5 .5]);
lnZmf = sum( q1*q2*th12 ) + entropy(q1) + entropy(q2);

g1 = sum(q2*th12, 2) );          % compute g1 = E_q2[ theta ]
q1 = normalize( exp( g1 ) );     % and update q(x1)
```

Model



$$f_{12} = \exp(\theta_1 x_1 + \theta_2 x_2 + \theta_{12} x_1 x_2)$$

$$\theta_1 = 0.5 \quad \theta_2 = 1.0 \quad \theta_{12} = 1.5$$

$$q_1 \propto \exp\Big[\sum_{x_2} q_2(x)\, \theta(x) \Big]$$

$$q_2 \propto \exp\Big[\sum_{x_1} q_1(x)\, \theta(x) \Big]$$

$$\log Z_{mf} \leq \log Z$$

2.511	3.237
2.695	

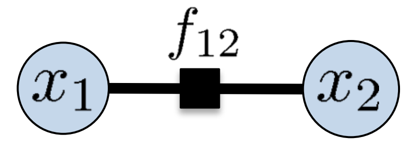
$q_1 :$ $\begin{array}{c|c} x_1 & \begin{pmatrix} 0.500 \\ 0.500 \end{pmatrix} \end{array}$

$q_2 :$ $\begin{array}{c} -x_2- \\ \begin{pmatrix} 0.500 & 0.500 \end{pmatrix} \end{array}$

$\log f_{12} :$ $\begin{array}{cc} & -x_2- \\ x_1 & \begin{pmatrix} 0.000 & 1.000 \\ 0.500 & 3.000 \end{pmatrix} \end{array}$

$$\begin{pmatrix} 0.500 \\ 1.750 \end{pmatrix} \rightarrow \begin{pmatrix} 0.223 \\ 0.777 \end{pmatrix}$$

Mean Field Example



Model

$$f_{12} = \exp(\theta_1 x_1 + \theta_2 x_2 + \theta_{12} x_1 x_2)$$

$$\theta_1 = 0.5 \quad \theta_2 = 1.0 \quad \theta_{12} = 1.5$$

```
th12 = factor([1 2], [0 1 ; 0.5  3 ]);% theta(x) = ...
f12 = exp(th12);                        % f(x) = exp( theta(x) )
lnZ = log(sum( f12 ));                  % ln Zf = 3.237

q1 = factor(1, [.5 .5]);          % initialize factored q(x)
q2 = factor(2, [.5 .5]);
lnZmf = sum( q1*q2*th12 ) + entropy(q1) + entropy(q2);

g2 = sum(q1*th12, 1) );           % compute g2 = E_q1[ theta ]
q2 = normalize( exp( g2 ) );      % and update q(x2)
```

$$q_1 \propto \exp\left[\sum_{x_2} q_2(x)\,\theta(x)\right]$$

$$q_2 \propto \exp\left[\sum_{x_1} q_1(x)\,\theta(x)\right]$$

$$\log Z_{mf} \leq \log Z$$

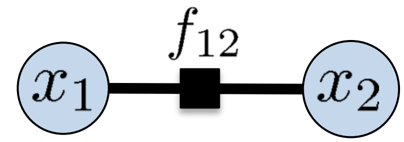
$\log Z_{mf}$	$\log Z$
2.511	3.237
2.695	
3.193	

$q_1:$ $\quad \begin{array}{c c} & \\ x_1 & \begin{bmatrix} 0.223 \\ 0.777 \end{bmatrix} \end{array}$ $\qquad q_2:$ $\begin{array}{c} -x_2- \\ \begin{bmatrix} 0.500 & 0.500 \end{bmatrix} \end{array}$

$\log f_{12}:$ $\begin{array}{c c} & -x_2- \\ x_1 & \begin{bmatrix} 0.000 & 1.000 \\ 0.500 & 3.000 \end{bmatrix} \end{array}$

$$\begin{bmatrix} 0.339 & 2.555 \end{bmatrix} \rightarrow \begin{bmatrix} 0.103 & 0.897 \end{bmatrix}$$

Mean Field Example

Model 

$$f_{12} = \exp(\theta_1 x_1 + \theta_2 x_2 + \theta_{12} x_1 x_2)$$

$$\theta_1 = 0.5 \quad \theta_2 = 1.0 \quad \theta_{12} = 1.5$$

```
th12 = factor([1 2], [0 1 ; 0.5  3 ]);% theta(x) = …
f12 = exp(th12);                       % f(x) = exp( theta(x) )
lnZ = log(sum( f12 ));                 % ln Zf = 3.237

q1 = factor(1, [.5 .5]);         % initialize factored q(x)
q2 = factor(2, [.5 .5]);
lnZmf = sum( q1*q2*th12 ) + entropy(q1) + entropy(q2);

g1 = sum(q2*th12, 2) );          % compute g1 = E_q2[ theta ]
q1 = normalize( exp( g1 ) );     % and update q(x1)
```

$$q_1 \propto \exp\Big[\sum_{x_2} q_2(x)\,\theta(x)\Big]$$

$$q_2 \propto \exp\Big[\sum_{x_1} q_1(x)\,\theta(x)\Big]$$

$$\log Z_{mf} \le \log Z$$

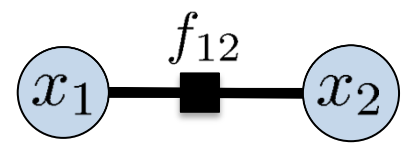
$\log Z_{mf}$	$\log Z$
2.511	3.237
2.695	
3.193	
3.221	

$q_1 :\ x_1 \begin{pmatrix} 0.223 \\ 0.777 \end{pmatrix}$ $\quad q_2 :\ \begin{array}{c} -x_2- \\ \begin{pmatrix} 0.103 & 0.897 \end{pmatrix} \end{array}$ $\quad \log f_{12} :\ \begin{array}{cc} & -x_2- \\ x_1 & \begin{pmatrix} 0.000 & 1.000 \\ 0.500 & 3.000 \end{pmatrix} \end{array}$

$$\begin{pmatrix} 0.897 \\ 2.743 \end{pmatrix} \rightarrow \begin{pmatrix} 0.136 \\ 0.864 \end{pmatrix}$$

Mean Field Example

Model



$$f_{12} = \exp(\theta_1 x_1 + \theta_2 x_2 + \theta_{12} x_1 x_2)$$

$$\theta_1 = 0.5 \quad \theta_2 = 1.0 \quad \theta_{12} = 1.5$$

```
th12 = factor([1 2], [0 1 ; 0.5  3 ]);% theta(x) = ...
f12 = exp(th12);                       % f(x) = exp( theta(x) )
lnZ = log(sum( f12 ));                 % ln Zf = 3.237

q1 = factor(1, [.5 .5]);        % initialize factored q(x)
q2 = factor(2, [.5 .5]);
lnZmf = sum( q1*q2*th12 ) + entropy(q1) + entropy(q2);

g1 = sum(q2*th12, 2) );         % compute g1 = E_q2[ theta ]
q1 = normalize( exp( g1 ) );    % and update q(x1)
```

$$q_1 \propto \exp\left[\sum_{x_2} q_2(x)\,\theta(x)\right]$$

$$q_2 \propto \exp\left[\sum_{x_1} q_1(x)\,\theta(x)\right]$$

q_1 : x_1 $\begin{pmatrix} 0.134 \\ 0.866 \end{pmatrix}$ $\quad$ q_2 : $-x_2-$ $\begin{pmatrix} 0.091 & 0.909 \end{pmatrix}$

$\log f_{12}$: $-x_2-$, x_1 $\begin{pmatrix} 0.000 & 1.000 \\ 0.500 & 3.000 \end{pmatrix}$

$\log Z_{mf}$	$\leq$	$\log Z$
2.511		3.237
2.695		
3.193		
3.221		
⋮		
3.222		

Coordinate ascent on q1, q2
Lower bound increases to (local) optimum

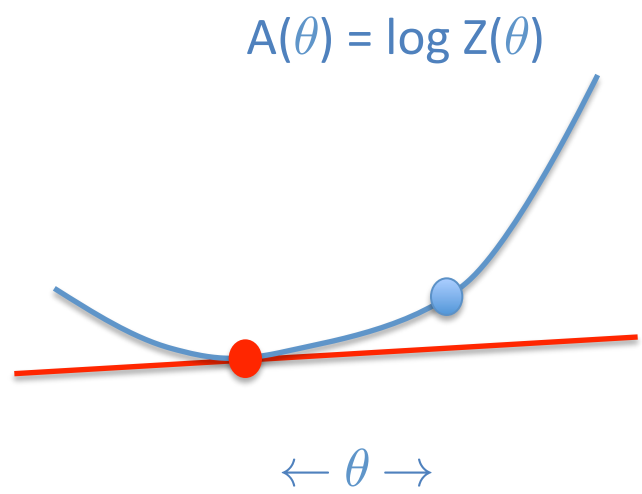
Geometry of mean field

- Suppose we want to evaluate A(θ) (convex)
- Choose a different θ' where A(θ') is easy:

$$\theta'(x) = \sum_i \theta_i(x_i) \quad \Leftrightarrow \quad q(x) = \prod q_i(x_i)$$

$$A(\theta') = \mathbb{E}_q[\theta'(x)] + H(q)$$
$$= \theta' \cdot \mu + H(\mu)$$

$$\nabla_\theta A(\theta') = \mu$$ where μ are the moments of q(x)



- We have a simple linear lower bound

$$A(\theta) \geq (\theta - \theta')\nabla A(\theta') + A(\theta')$$
$$= (\theta - \theta') \cdot \mu + \theta' \cdot \mu + H(\mu)$$
$$= \theta \cdot \mu + H(\mu)$$ where μ are the moments of the "easy" q(x)

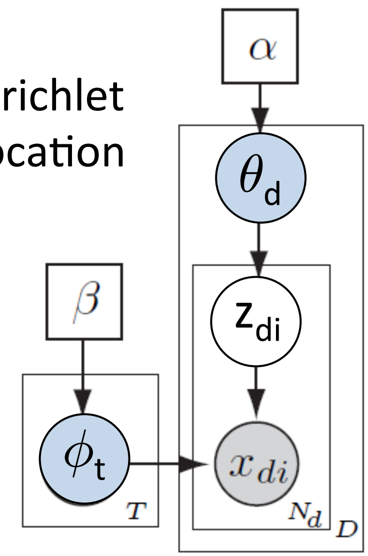
- Coordinate ascent: optimize q(x) = optimize choice of θ'

Using mean field

- Mean field methods are often used in (variational) EM

$$\max_{\theta} \max_{q} \; -D\big(\hat{p}(x)q(z|x) \| p(x,z\,;\,\theta)\big)$$

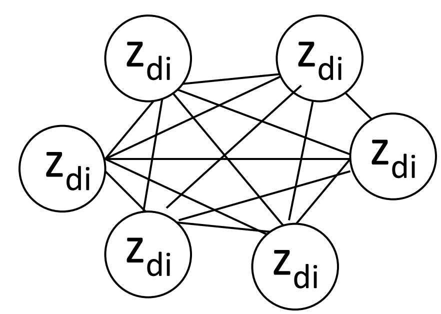
Latent Dirichlet Allocation



- E-step: max with respect to q
  - Optimal q(z) = p(z | x, θ)
- M-step: max with respect to θ
  - Expected complete log-likelihood

- Example: Collapsed LDA

$$p(z_{di} = t | \mathbf{z}^{\neg di}, \mathbf{x}, \alpha, \beta) \propto \alpha + N_{dt}^{\neg di} \; \frac{\beta + N_{wt}^{\neg di}}{W\beta + N_t^{\neg di}}$$



Mean field updates are (something like)

$$q(z_{di}) \propto \exp\left[\log(\alpha + \mathbb{E}_q[N_{dt}^{\neg di}]) + \; \log\Big(\frac{\beta + \mathbb{E}_q[N_{wt}^{\neg di}]}{W\beta + \mathbb{E}_q[N_t^{\neg di}]}\Big)\right]$$

Duality and the log-partition function

- Represent LPF as optimization over distributions

$$\log Z = \max_{q \in \mathbb{P}} \; \mathbb{E}_q[\log f(x)] + H(x\,;\,q)$$

(maximum: $q = p$)

- Two problems:
  - Constraint set is too complex (as in LP form)
  - Entropy defined in terms of q, not available

- Proof: $$D(q \| p) = \sum_x q(x) \log \left[\frac{q(x)}{\frac{1}{Z} f(x)} \right]$$

(Kullback–Leibler divergence)

$$= -H(x\,;\,q) - \mathbb{E}_q[\log f(x)] + \log Z$$

$$\Rightarrow \log Z \geq \mathbb{E}_q[\log f(x)] + H(x\,;\,q)$$

equal iff $p = q$

Entropy approximations

- Approximate H(x;q) with something "easier to evaluate"
  - Typically, defined solely in terms of low-order moments
  - If we relax M to L, then our approximate H(.) must be defined in all of L

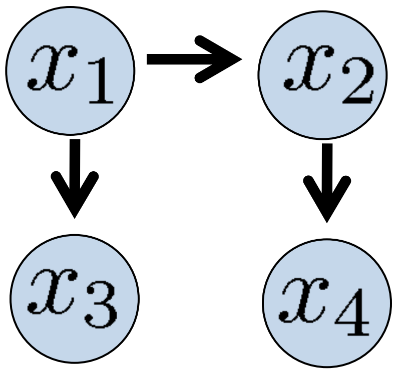
Bethe approximation

$$H(x) \approx \sum_i H_i - \sum_{(ij)\in E} I_{ij} \stackrel{\text{def}}{=} \hat{H}_{bethe}$$

where $H_i = -\sum_x b_i(x_i)\log(b_i(x))$

$$I_{ij} = \sum_x b_{ij}(x_i, x_j)\log\left(\frac{b_{ij}(x_i, x_j)}{b_i(x_i)b_j(x_j)}\right)$$

Entropy in trees



$$p(x_1^i)\, p(x_2^i | x_1^i) = p(x_2^i)\, p(x_1^i | x_2^i) = p(x_1^i)\, p(x_2^i)\, \frac{p(x_1^i, x_2^i)}{p(x_1^i)\, p(x_2^i)}$$

Then, $$\mathbb{H} = \sum_i H[p(x_i)] - \sum_{ij} \mathbb{I}[p(x_i, x_j)]$$

This is essentially the same trick used in the Chow-Liu learning algorithm!

We can use this as an approximation for graphs with cycles

Called the “Bethe” approximation (see e.g., Yedidia et al. 2001)

Bethe Approximation

- Suppose we want to optimize

$$\max_{b \in \mathbb{L}} \sum_{\alpha} \mathbb{E}_{b_\alpha}[\theta_\alpha(x_\alpha)] + + \sum_i \mathbb{H}(b_i) - \sum_{ij} \mathbb{I}(b_{ij})$$

$$\mathbb{L}_G = \{b_i, b_{ij} : \sum_{x_i} b_{ij}(x_i, x_j) = b_j(x_j), \sum_{x_j} b_j(x_j) = 1, b_{ij} \geq 0\}$$

- Lagrange multipliers…
  - On board

Loopy BP and the partition function

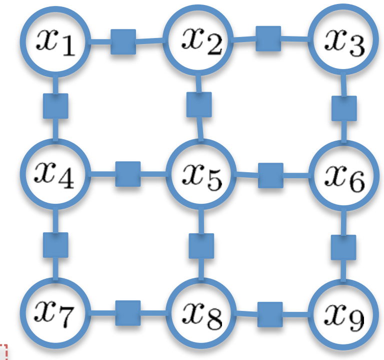
- Use the Bethe approximation to estimate log Z:
  - Run loopy BP on the factor graph & calculate beliefs
  - Use the Bethe approximation to H(b):

$$\log Z \approx \sum_{\alpha} \mathbb{E}_{b_\alpha}[\log f_\alpha(x_\alpha)] + \sum_i \mathbb{H}(b_i) - \sum_\alpha \mathbb{E}_{b_\alpha}\left[\log \frac{b_\alpha}{\prod_{i\in\alpha} b_i}\right]$$

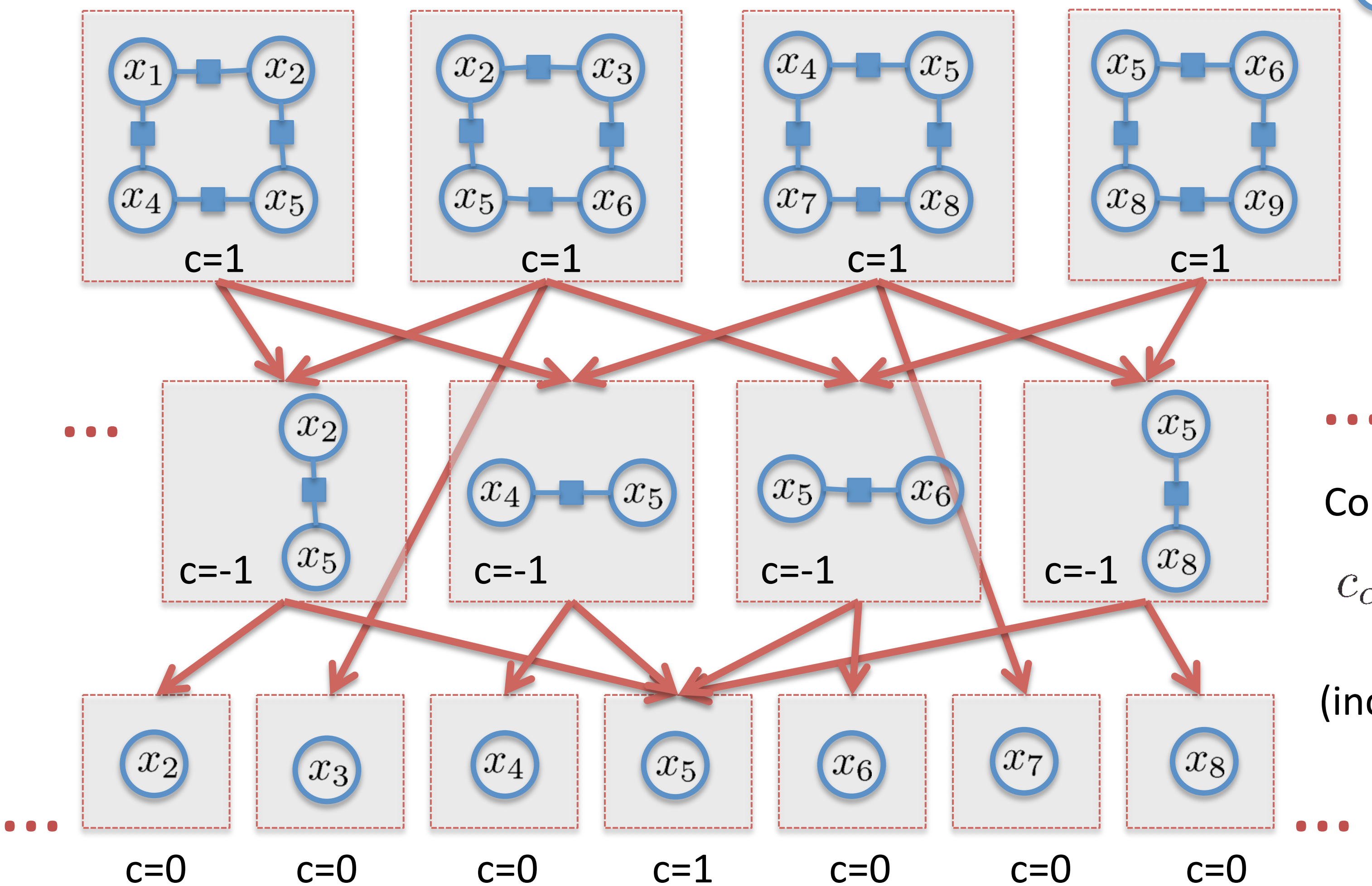
  - Often written using counting numbers: $c_\alpha = 1, \quad c_i = 1 - \deg(i)$

$$\log Z \approx \sum_{\alpha} \mathbb{E}_{b_\alpha}[\log f_\alpha(x_\alpha)] + \sum_\alpha c_\alpha H(b_\alpha)$$

Region graphs



Region: a collection of variables & their interactions



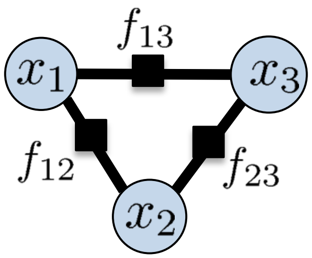
Counting numbers:

$$c_\alpha = 1 - \sum_{\beta \supset \alpha} c_\beta$$

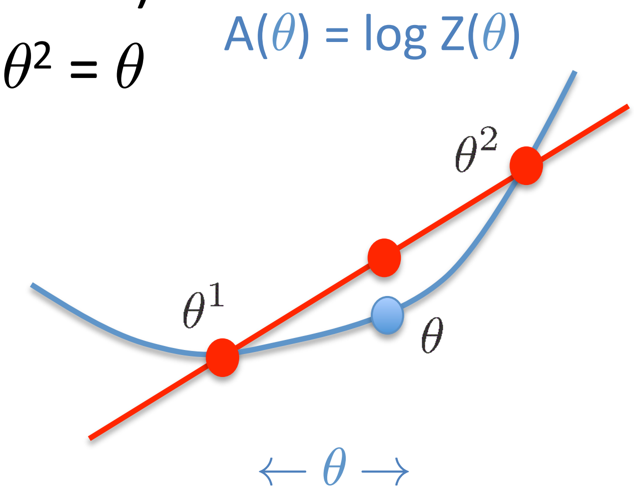
(inclusion/exclusion)

Geometry of tree-reweighted BP

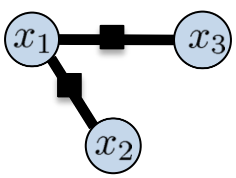
- Suppose we want to evaluate A(θ) (convex)
- Choose easy θ^1 , θ^2 , where wθ^1 + (1-w)θ^2 = θ



$$\log f(x) = \theta(x) = \sum \theta_{ij}(x_i, x_j)$$

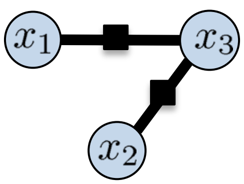


$$\theta_{12}: \begin{array}{cc} & x_2 \\ x_1 & \begin{bmatrix} 1.0 & 2.0 \\ 0.0 & 1.0 \end{bmatrix} \end{array} \qquad \theta_{13}: \begin{array}{cc} & x_3 \\ x_1 & \begin{bmatrix} -1.0 & 0.0 \\ -1.0 & 1.0 \end{bmatrix} \end{array} \qquad \theta_{23}: \begin{array}{cc} & x_3 \\ x_2 & \begin{bmatrix} -1.0 & 0.0 \\ 2.0 & 1.0 \end{bmatrix} \end{array}$$



$$\theta^1(x) = 2\,\theta_{12}(x) + \theta_{13}(x)$$

A(θ^1) = 4.707



$$\theta^2(x) = \theta_{13}(x) + 2\,\theta_{23}(x)$$

A(θ^2) = 4.269

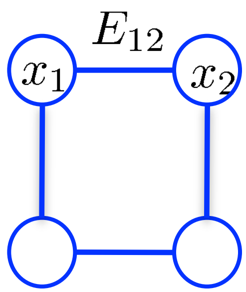
$$\theta(x) = .5\,\theta^1(x) + .5\theta^2(x)$$

$$A(\theta) \leq .5\,A(\theta^1) + .5\,A(\theta^2)$$

A(θ) = 4.298 $\leq$ (4.707 + 4.269)/2 = 4.488

EXTRAS

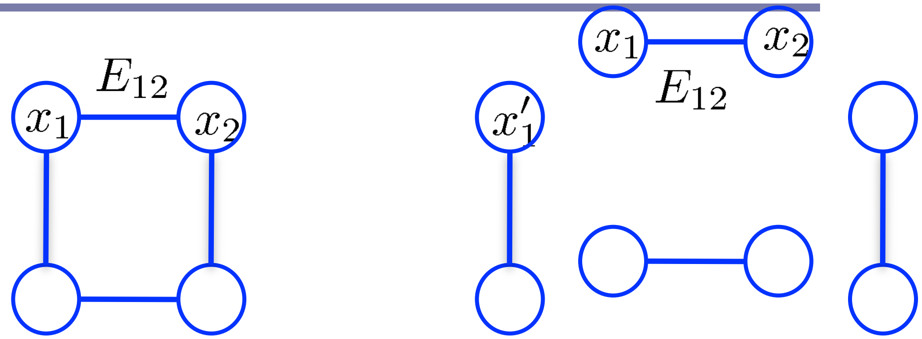
Dual decomposition



Original

$$\max_{\underline{\mathrm{x}}} \sum_{ij} E_{ij}(x_i, x_j)$$

Dual decomposition

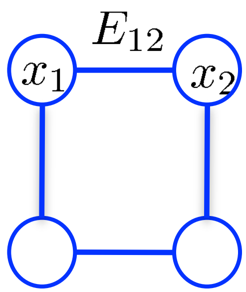


Original **Decomposition**

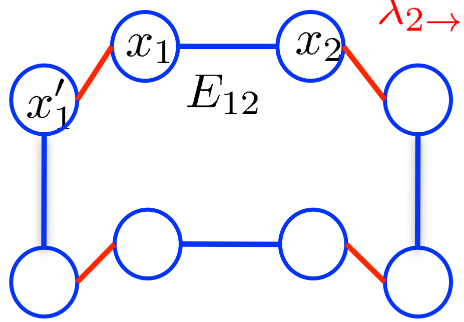
$$\max_{\underline{x}} \sum_{ij} E_{ij}(x_i, x_j) \leq \sum_{ij} \max_{\underline{x}} E_{ij}(x_i, x_j)$$

- Decompose graph into smaller subproblems
- Solve each independently; optimistic bound
- Exact if all copies agree

Dual decomposition



Original



Decomposition

$\lambda_{2\to12}(x_2)$

$\lambda_{2\to24}(x_2)$
$= -\lambda_{2\to12}(x_2)$

Add factors that “adjust” each local term, but cancel out in total

Total addition for xi is zero:

$$\forall i \sum_j \lambda_{i\to ij}(x_i) = 0$$

$$\max_{\underline{\mathrm{x}}} \sum_{ij} E_{ij}(x_i, x_j) \le \min_{\lambda} \sum_{ij} \max_{\underline{\mathrm{x}}} E_{ij}(x_i, x_j) + \lambda_{i\to ij}(x_i) + \lambda_{j\to ij}(x_j)$$

- Decompose graph into smaller subproblems
- Solve each independently; optimistic bound
- Exact if all copies agree
- Enforce lost equality constraints via Langrange multipliers

Duality relationship

- These views are Lagrangian duals:

$$\log f(x^*) \le \max_{\mu} \Big[\sum_{i,k} \theta_{i;k}\, \mu_{i;k} + \sum_{i,j,k,l} \theta_{ij;kl}\, \mu_{ij;kl} \Big]$$

subject to (a) normalization constraints (enforce explicitly)

(b) consistency: $\sum_l \mu_{ij;kl} = \mu_{i;k}$, $\sum_k \mu_{ij;kl} = \mu_{j;l}$ (use Lagrange)

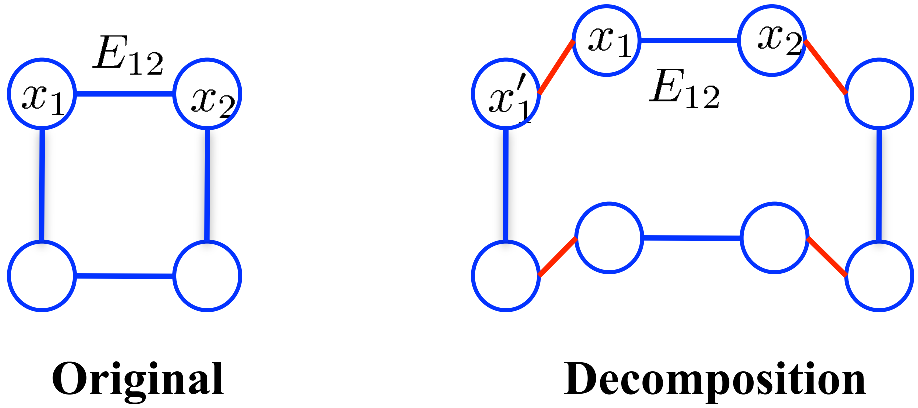
$$L = \max_{\mu} \min_{\lambda} \sum_{i,k} \theta_{i;k}\, \mu_{i;k} + \sum_{i,j,k,l} \theta_{ij;kl}\, \mu_{ij;kl} + \sum_{i,j,k} \lambda_{i\to ij;k} \Big(\sum_{l} \mu_{ij;kl} - \mu_{i;k} \Big)$$

$$\le \min_{\lambda} \max_{\mu} \sum_{i,k} \theta_{i;k}\, \mu_{i;k} + \sum_{i,j,k,l} \theta_{ij;kl}\, \mu_{ij;kl} + \sum_{i,j,k} \lambda_{i\to ij;k} \Big(\sum_{l} \mu_{ij;kl} - \mu_{i;k} \Big)$$

$$= \min_{\lambda} \max_{\mu} \sum_{i,k} \Big(\theta_{i;k} - \sum_{j} \lambda_{i\to ij;k}\Big) \mu_{i;k} + \sum_{i,j,k,l} \big(\theta_{ij;kl} + \lambda_{i\to ij;k} + \lambda_{j\to ij;l}\big) \mu_{ij;kl}$$

$$= \min_{\lambda} \sum_{i,k} \max_{k} \Big(\theta_{i;k} - \sum_{j} \lambda_{i\to ij;k}\Big) + \sum_{i,j,k,l} \max_{k,l} \big(\theta_{ij;kl} + \lambda_{i\to ij;k} + \lambda_{j\to ij;l}\big)$$

Decomposition view



Total addition for xi is zero:

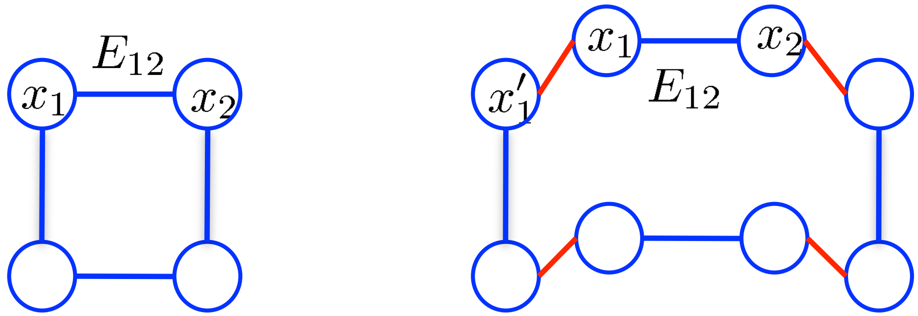
$$\forall i \sum_j \lambda_{i \to ij}(x_i) = 0$$

$$\max_{\underline{x}} \sum_{ij} E_{ij}(x_i, x_j) \le \min_{\lambda} \sum_{ij} \max_{\underline{x}} E_{ij}(x_i, x_j) + \lambda_{i \to ij}(x_i) + \lambda_{j \to ij}(x_j)$$

Many approaches to optimizing this bound

- Dual decomposition (Komodakis et al. 2007)
- TRW, MPLP (Wainwright et al. 2005; Globerson & Jaakkola 2007)
- Soft arc consistency (Cooper & Schiex 2004)

Decomposition view

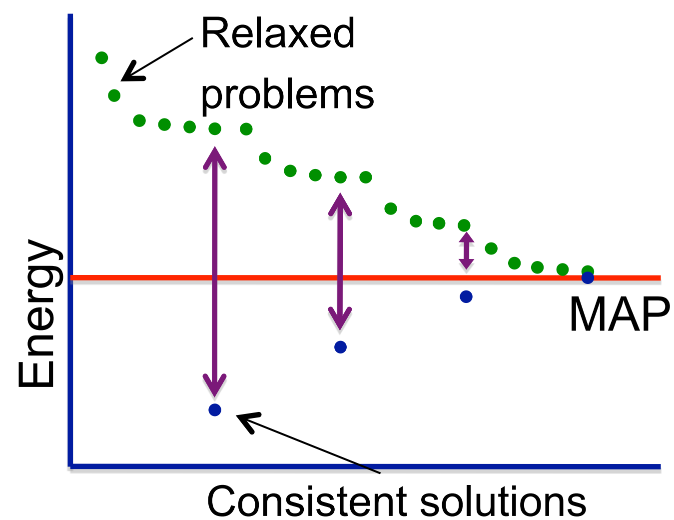


Original **Decomposition**

Total addition for xi is zero:

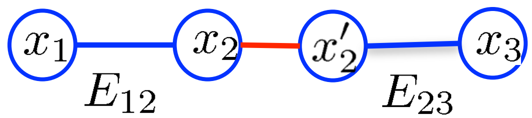
$$\forall i \sum_j \lambda_{i\to ij}(x_i) = 0$$

$$\max_{\underline{\mathrm{x}}} \sum_{ij} E_{ij}(x_i, x_j) \leq \min_{\lambda} \sum_{ij} \max_{\underline{\mathrm{x}}} E_{ij}(x_i, x_j) + \lambda_{i\to ij}(x_i) + \lambda_{j\to ij}(x_j)$$



Optimizing the bound

$$\max_{\underline{x}} \sum_{ij} E_{ij}(x_i, x_j) \le \min_{\lambda} \sum_{ij} \max_{\underline{x}} E_{ij}(x_i, x_j) + \lambda_{i\to ij}(x_i) + \lambda_{j\to ij}(x_j)$$



$$\forall i \sum_j \lambda_{i\to ij}(x_i) = 0$$

x_2 \ $\leftarrow x_1 \rightarrow$			
↑	1	0	1
x_2	0	1	0
↓	1	2	1

λ

0	0
0	0
0	0

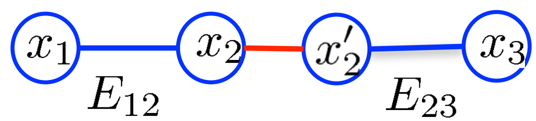
$\leftarrow x_3 \rightarrow$			x'_2
5	2	2	↑
1	2	0	x'_2
0	0	2	↓

Subgradient descent

- Find each subproblem's optimal configuration
- Adjust entries for mis-matched solutions

Optimizing the bound

$$\max_{\underline{\mathrm{x}}} \sum_{ij} E_{ij}(x_i, x_j) \le \min_{\lambda} \sum_{ij} \max_{\underline{\mathrm{x}}} E_{ij}(x_i, x_j) + \lambda_{i \to ij}(x_i) + \lambda_{j \to ij}(x_j)$$



$$\forall i \sum_j \lambda_{i \to ij}(x_i) = 0$$

$\leftarrow x_1 \rightarrow$ (columns), $\uparrow x_2 \downarrow$ (rows):

1	0	1
0	1	0
1	**2**	1

λ

0	0
0	0
0	0

$\leftarrow x_3 \rightarrow$ (columns), $\uparrow x_2' \downarrow$ (rows):

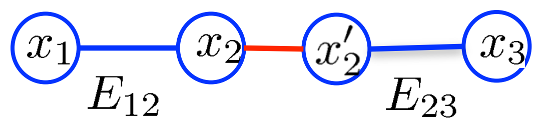
5	2	2
1	2	0
0	0	2

Subgradient descent

- Find each subproblem's optimal configuration
- Adjust entries for mis-matched solutions

Optimizing the bound

$$\max_{\underline{x}} \sum_{ij} E_{ij}(x_i, x_j) \le \min_{\lambda} \sum_{ij} \max_{\underline{x}} E_{ij}(x_i, x_j) + \lambda_{i\to ij}(x_i) + \lambda_{j\to ij}(x_j)$$



$$\forall i \sum_j \lambda_{i\to ij}(x_i) = 0$$

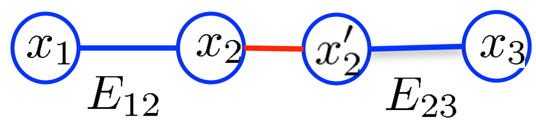
	$\leftarrow x_1 \rightarrow$			λ		$\leftarrow x_3 \rightarrow$			
$\uparrow$	2	1	2	+1	-1	4	1	1	$\uparrow$
x_2	0	1	0	0	0	1	2	0	x_2'
$\downarrow$	0	1	0	-1	+1	1	1	3	$\downarrow$

Subgradient descent

- Find each subproblem's optimal configuration
- Adjust entries for mis-matched solutions

Optimizing the bound

$$\max_{\underline{\mathrm{x}}} \sum_{ij} E_{ij}(x_i, x_j) \le \min_{\lambda} \sum_{ij} \max_{\underline{\mathrm{x}}} E_{ij}(x_i, x_j) + \lambda_{i \to ij}(x_i) + \lambda_{j \to ij}(x_j)$$



$$\forall i \sum_{j} \lambda_{i \to ij}(x_i) = 0$$

E_{12}: rows x_2 (↓), columns x_1 (→)

1	0	1
0	1	0
1	2	1

λ

0	0
0	0
0	0

E_{23}: rows x_2' (↓), columns x_3 (→)

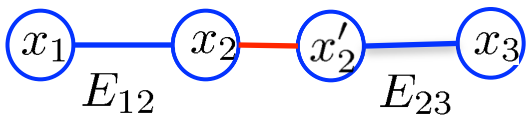
5	2	2
1	2	0
0	0	2

Coordinate descent

- Find lambda to optimize a sub-part of the bound
- Lots of equivalent choices
- "Max-sum diffusion" $\lambda_{23}(x_2) = -\lambda_{21}(x_2) = \frac{1}{2}\left(\max_{x_1} \tilde{E}_{12}(x_1, x_2) - \max_{x_3} \tilde{E}_{23}(x_2, x_3)\right)$

Optimizing the bound

$$\max_{\underline{x}} \sum_{ij} E_{ij}(x_i, x_j) \le \min_{\lambda} \sum_{ij} \max_{\underline{x}} E_{ij}(x_i, x_j) + \lambda_{i \to ij}(x_i) + \lambda_{j \to ij}(x_j)$$



$$\forall i \sum_j \lambda_{i \to ij}(x_i) = 0$$

$\leftarrow x_1 \rightarrow$ (columns), $\uparrow x_2 \downarrow$ (rows)

3	**2**	**3**
.5	**1.5**	**.5**
1	**2**	**1**

λ

2	**-2**
.5	**-.5**
0	**0**

$\leftarrow x_3 \rightarrow$ (columns), $\uparrow x_2' \downarrow$ (rows)

3	**0**	**0**
.5	**1.5**	**0**
0	**0**	**2**

Coordinate descent

- Find lambda to optimize a sub-part of the bound
- Lots of equivalent choices
- "Max-sum diffusion" $\lambda_{23}(x_2) = -\lambda_{21}(x_2) = \frac{1}{2}\left(\max_{x_1} \tilde{E}_{12}(x_1, x_2) - \max_{x_3} \tilde{E}_{23}(x_2, x_3)\right)$

Fixed-point updates

- Simple fixed-point algorithm for tightening the dual

While ~ done
  for i = 1…n

$$\forall \alpha \ni i: \;\; \gamma_\alpha(x_i) = \max_{\alpha \setminus i} \theta_\alpha(x_\alpha)$$
(Find max-marginals of all factors with i)

$$\bar{\gamma}(x_i) = \frac{1}{d_\alpha} \sum_\alpha \gamma_\alpha(x_i)$$
(Compute their average)

$$\hat{x}_i = \arg\max_{x_i} \bar{\gamma}(x_i)$$
(Guess a good value for xi)

$$\forall \alpha \ni i: \;\; \theta_\alpha(x_\alpha) \leftarrow \theta_\alpha(x_\alpha) - \gamma_\alpha(x_i) + \bar{\gamma}(x_i)$$
(Update all factors to match)

  end;
  Upper bound: $\theta^+ = \sum_\alpha \max_{x_\alpha} \theta_\alpha(x_\alpha)$

  Lower bound: if $\theta(\hat{x}) \geq \theta(x^*)$, set $x^* = \hat{x}$ *(Save best config so far)*
end;